Logical Foundations of Services

Ionuţ Ţuţu[1,2]

1 **Department of Computer Science, Royal Holloway University of London**
2 **Institute of Mathematics of the Romanian Academy, Research group of the project ID-3-0439**
ittutu@gmail.com

Abstract

In this paper we consider a logical system of networks of processes that interact in an asynchronous manner by exchanging messages through communication channels. This provides a foundational algebraic framework for service-oriented computing that constitutes a primary factor in defining logical specifications of services, the way models of these specifications capture service orchestrations, and how properties of interaction-points, i.e. points through which such networks connect to one another, can be expressed. We formalise the resulting logic as a parameterised institution, which promotes the development of both declarative and operational semantics of services in a heterogeneous setting by means of logic-programming concepts.

1998 ACM Subject Classification F.4.1 Mathematical Logic

Keywords and phrases Formal methods, Service-oriented computing, Institution theory

Digital Object Identifier 10.4230/OASIcs.ICCSW.2013.111

1 Introduction

Service-oriented computing is a recent paradigm focusing on computation in data processing infrastructures that are globally available, and in which software applications can discover and bind dynamically to services offered by providers. The present paper builds on earlier theoretical work on algebraic structures that capture the way services are orchestrated [8, 6], and on the mechanisms that formalise the discovery of services that can be bound to a client application in terms of logical specifications of required/provided services [9, 7]. It explores one of the most technical aspects of a recent approach proposed in [3] that describes how aspects specific to the logic-programming paradigm can be used to capture the declarative and operational semantics of service-oriented computing.

To this purpose, we advance an integrated algebraic framework that constitutes the primary factor in defining logical specifications of services, as well as models of these specifications that correspond to orchestrations of components depending upon externally provided services. Our work upgrades the formalism considered in [3] by making a clear distinction between specifications of services and their orchestrations, which results in a framework that we consider to be more appropriate for addressing aspects such as heterogeneity.

Since the logic-programming semantics of services is technically based on an underlying logical system that is formalised as an institution [10], we will concentrate our efforts on proving that the proposed logic of networks of processes constitutes an institution. We recall that the theory of institutions is a categorical abstract model theory [4] that promotes a universal approach to the study of logics by abstracting the notion of truth, which is supposed to be invariant with respect to the change of notation. Formally, an *institution* is a quadruple $\mathcal{I} = \langle \mathbb{S}\mathrm{ig}^{\mathcal{I}}, \mathsf{Sen}^{\mathcal{I}}, \mathsf{Mod}^{\mathcal{I}}, \models^{\mathcal{I}} \rangle$ that consists of

– a category $\mathbb{S}\mathrm{ig}^{\mathcal{I}}$ of *signatures* and *signature morphisms*,

2013 Imperial College Computing Student Workshop (ICCSW'13).
Editors: Andrew V. Jones, Nicholas Ng; pp. 111–118

OpenAccess Series in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

- a *sentence functor* $\mathsf{Sen}^{\mathcal{I}}\colon \mathbb{S}\mathrm{ig}^{\mathcal{I}} \to \mathbb{S}\mathrm{et}$ defining for every signature Σ the set $\mathsf{Sen}^{\mathcal{I}}(\Sigma)$ of Σ-*sentences*, and for every signature morphism $\varphi\colon \Sigma \to \Sigma'$ the *sentence translation map* $\mathsf{Sen}^{\mathcal{I}}(\varphi)\colon \mathsf{Sen}^{\mathcal{I}}(\Sigma) \to \mathsf{Sen}^{\mathcal{I}}(\Sigma')$,
- a *model functor* $\mathsf{Mod}^{\mathcal{I}}\colon (\mathbb{S}\mathrm{ig}^{\mathcal{I}})^{op} \to \mathbb{C}\mathrm{at}$ defining for every signature Σ the category $\mathsf{Mod}^{\mathcal{I}}(\Sigma)$ of Σ-*models* and Σ-*homomorphisms*, and for every morphism $\varphi\colon \Sigma \to \Sigma'$, the *reduct functor* $\mathsf{Mod}^{\mathcal{I}}(\varphi)\colon \mathsf{Mod}^{\mathcal{I}}(\Sigma') \to \mathsf{Mod}^{\mathcal{I}}(\Sigma)$,
- a family of *satisfaction* relations $\models^{\mathcal{I}}_{\Sigma} \subseteq |\mathsf{Mod}^{\mathcal{I}}(\Sigma)| \times \mathsf{Sen}^{\mathcal{I}}(\Sigma)$, indexed by signatures,

such that the following *satisfaction condition* holds:

$$M' \models^{\mathcal{I}}_{\Sigma'} \mathsf{Sen}^{\mathcal{I}}(\varphi)(\rho) \quad \text{if and only if} \quad \mathsf{Mod}^{\mathcal{I}}(\varphi)(M') \models^{\mathcal{I}}_{\Sigma} \rho,$$

for every signature morphism $\varphi\colon \Sigma \to \Sigma'$, Σ'-model M' and Σ-sentence ρ.

2 Asynchronous Relational Networks

The first step towards the formulation of the logical framework for service orchestrations is the introduction of a parameterised construction of a category $\langle \mathbb{S}\mathrm{ig}, L\rangle$-$\mathbb{ARN}$ of asynchronous relational networks. In the subsequent sections of the present paper this will serve as foundation for both the syntactic and the semantic dimensions of the proposed logical system.

Throughout this section we will consider

- a fixed category $\mathbb{S}\mathrm{ig}$ (of *signatures* and *signature morphisms*) and
- a *behavioural labelling functor* $L\colon |\mathbb{S}\mathrm{ig}| \to \mathbb{C}\mathrm{at}$ assigning to every signature $\Sigma \in |\mathbb{S}\mathrm{ig}|$ a category $L(\Sigma)$ of *behavioural* Σ-*labels*.

For presentation purposes, we will assume $\mathbb{S}\mathrm{ig}$ to be the category of signatures of linear temporal logic, and $L(\Sigma)$ to be the category of Σ-presentations, i.e. of sets of Σ-sentences.

The asynchronous relational networks that we examine here follow closely the formalisation from [3], and are similar to those defined in [6] except that we rely on hypergraphs [5, 2] instead of graphs, and on abstract labels instead of signatures of linear temporal logic and sets of traces. Just as in [6], the proposed concepts reflect the intuitive representation of service components as networks of processes that interact asynchronously by exchanging messages through communication channels. However, since the role of messages (organised into structures called ports) is merely to provide a convenient description of certain signatures of linear temporal logic, we choose not to introduce them explicitly, but to refer directly to the signatures they designate, similarly to [9].

Processes are defined by sets of interaction-points labelled with port signatures and by behavioural labels that correspond to the way the processes operate.

▶ **Definition 1** (Process). A $\langle \mathbb{S}\mathrm{ig}, L\rangle$-*process* $\langle X, (\Sigma_x \xrightarrow{\iota_x} \Sigma)_{x \in X}, \Lambda\rangle$ consists of a set X of *interaction-points*, each point $x \in X$ being labelled with a *port signature* $\Sigma_x \in |\mathbb{S}\mathrm{ig}|$, a *process signature* $\Sigma \in |\mathbb{S}\mathrm{ig}|$ such that $(\Sigma_x \xrightarrow{\iota_x} \Sigma)_{x \in X}$ is a coproduct in $\mathbb{S}\mathrm{ig}$, and a label $\Lambda \in |L(\Sigma)|$.

In the actual situations the port signatures Σ_x are sets of actions that match either the *publication* $m!$ of an outgoing message m or the *delivery* $m¡$ of an incoming message m. The process signature Σ is given by the disjoint union $\biguplus_{x \in X} \Sigma_x = \{x.a \mid x \in X, a \in \Sigma_x\}$, while the injections ι_x are the obvious prefix maps $x.\_\colon \Sigma_x \to \Sigma$. The behaviour of the process is specified through a set Λ of sentences of linear temporal logic over Σ.

▶ **Example 2.** In Figure 1a we depict a process JourneyPlanner that provides directions from a source to a target location. The process interacts with the environment through two interaction-points: JP_1 and JP_2. The first one is used for communicating with potential

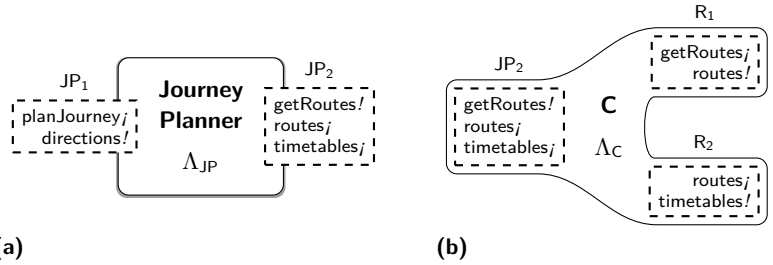

Figure 1 The ARN JourneyPlanner: (a) the process JourneyPlanner, (b) the connection C.

client processes – the request for directions (including the source and the target locations) is encoded into the delivery action planJourney¡, while the response is represented by the publication action directions!. The second point defines actions that correspond to the interaction between JourneyPlanner and other necessary processes – the publication action getRoutes! can be seen as a query for all possible routes between the specified source and target locations, while routes¡ and timetables¡ define the delivery of the result of the query and of the timetables of the available transport services for the selected routes.

The behaviour of the process JourneyPlanner can be described as follows:

– Whenever it receives a request planJourney¡ it immediately initiates the search of available routes through the publication action getRoutes!.

$$\Box\,(\mathsf{planJourney}_{¡} \supset \bigcirc\,\mathsf{getRoutes}!)$$

– Once it receives both the routes and the corresponding timetables it compiles the directions and replies to the client.

$$\Box\,\Big(\mathsf{getRoutes}! \supset \big(\mathsf{routes}_{¡}\,\mathcal{R}\,(\mathsf{routes}_{¡} \supset (\mathsf{timetables}_{¡}\,\mathcal{R}\,(\mathsf{timetables}_{¡} \supset \Diamond\,\mathsf{directions}!))) \vee \mathsf{timetables}_{¡}\,\mathcal{R}\,(\mathsf{timetables}_{¡} \supset (\mathsf{routes}_{¡}\,\mathcal{R}\,(\mathsf{routes}_{¡} \supset \Diamond\,\mathsf{directions}!))))\big)\Big)$$

Processes communicate by transmitting messages through channels. As in [1, 6], channels are bidirectional, in the sense that they may transmit both incoming and outgoing messages.

▶ **Definition 3** (Channel). A $\langle \mathbb{S}\mathrm{ig}, L\rangle$-*channel* $\langle \Sigma, \Lambda\rangle$ consists of a *channel signature* $\Sigma \in |\mathbb{S}\mathrm{ig}|$ and a behavioural label $\Lambda \in |L(\Sigma)|$.

Note that channels do not provide any information about the interacting entities, but only on how the communication is realised. In order to enable the communication between given processes, channels need to be attached to their interaction-points, thus forming connections.

▶ **Definition 4** (Connection). A $\langle \mathbb{S}\mathrm{ig}, L\rangle$-*connection* $\langle \Sigma, \Lambda, (\Sigma \xleftarrow{\iota_x} \Sigma'_x \xrightarrow{\mu_x} \Sigma_x)_{x\in X}\rangle$ between the port signatures $(\Sigma_x)_{x\in X}$, where X is a set of *interaction-points*, consists of a channel $\langle \Sigma, \Lambda\rangle$ and a family of *attachment spans* $(\Sigma \xleftarrow{\iota_x} \Sigma'_x \xrightarrow{\mu_x} \Sigma_x)_{x\in X}$ in $\mathbb{S}\mathrm{ig}$.

The channel signature Σ is given in general by a set of both publication and delivery actions $m!$ and $m_¡$ determined by messages m. As in the case of processes, the behaviour of the channel is specified through a set of sentences of linear temporal logic over Σ. The maps ι_x and μ_x are considered to be set-theoretic inclusions and polarity-preserving injections, respectively, i.e. injective functions μ_x with the property that $\mu_x(a)$ is a publication action of Σ_x if and only if a is a publication action of Σ, and for this reason they are often presented as

partial maps $\mu_x \colon \Sigma \rightharpoonup \Sigma_x$; in addition, the signatures $(\Sigma'_x)_{x \in X}$ are usually chosen such that for any point $x \in X$, $m! \in \Sigma'_x$ ($m¿ \in \Sigma'_x$) if and only if $m¿ \in \Sigma'_y$ ($m! \in \Sigma'_y$) for some $y \in X \setminus \{x\}$. This condition ensures an appropriate pairing of messages: every published message of Σ_x, for $x \in X$, is paired with a delivered message of Σ_y, for some $y \in X \setminus \{x\}$, and vice versa.

▶ **Example 5.** In order to illustrate how the process JourneyPlanner can interact with other processes, we consider the connection C depicted in Figure 1b that moderates the flow of messages between the port signature named JP_2 and two other port signatures, R_1 and R_2.

The underlying channel of C is given by the set of actions $\Sigma = \{g!, g¿, r!, r¿, t!, t¿\}$ together with the set of sentences of linear temporal logic

$$\Lambda_{\mathsf{C}} = \left\{ \Box \left(m! \to \bigcirc m¿ \right) \mid m \in \{g, r, t\} \right\}$$

that specifies the delivery of all published messages without any delay. It is attached to the port signatures JP_2, R_1 and R_2 through the partial injections μ_{JP_2}, μ_{R_1} and μ_{R_2} given by

- $\mu_{\mathsf{JP}_2} = \{g! \mapsto \mathsf{getRoutes}!, r¿ \mapsto \mathsf{routes}¿, t¿ \mapsto \mathsf{timetables}¿\}$,
- $\mu_{\mathsf{R}_1} = \{g¿ \mapsto \mathsf{getRoutes}¿, r! \mapsto \mathsf{routes}!\}$ and
- $\mu_{\mathsf{R}_2} = \{r¿ \mapsto \mathsf{routes}¿, t! \mapsto \mathsf{timetables}!\}$.

Note that the actual senders and receivers of messages are specified through the attachment injections. For example, the message g is delivered only to the port signature R_1 (because μ_{R_2} is not defined on $g¿$), while r is simultaneously delivered to both JP_2 and R_2.

We can now define asynchronous networks of processes as hypergraphs having vertices labelled with port signatures and hyperedges labelled with processes and connections.

▶ **Definition 6** (Hypergraph)**.** An *(edge-labelled) hypergraph* $\langle X, E, \gamma \rangle$ consists of a set X of *vertices* or *nodes*, a set E of *hyperedges* disjoint from X, and an *incidence map* $\gamma \colon E \to \mathcal{P}(X)$ defining for every hyperedge $e \in E$ a non-empty set $\gamma_e \subseteq X$ of vertices it is incident with.

A hypergraph $\langle X, E, \gamma \rangle$ is said to be *edge-bipartite* if E is partitioned into two subsets F and G such that no adjacent hyperedges belong to the same partition, i.e. for every two hyperedges $e_1, e_2 \in E$ such that $\gamma_{e_1} \cap \gamma_{e_2} \neq \emptyset$, either $e_1 \in F$ and $e_2 \in G$, or $e_1 \in G$ and $e_2 \in F$.

▶ **Definition 7** (Asynchronous Relational Network – ARN)**.** A $\langle \mathbb{S}\mathrm{ig}, L \rangle$*-asynchronous relational network* $A = \langle X, P, C, \gamma, \Sigma, \iota, \mu, \Lambda \rangle$ consists of an edge-bipartite hypergraph $\langle X, P, C, \gamma \rangle$ of *points* $x \in X$, *computation hyperedges* $p \in P$, *communication hyperedges* $c \in C$, together with

- a *port signature* $\Sigma_x \in |\mathbb{S}\mathrm{ig}|$ for every point $x \in X$,
- a process $\left\langle \gamma_p, \left(\Sigma_x \xrightarrow{\iota^p_x} \Sigma_p \right)_{x \in \gamma_p}, \Lambda_p \right\rangle$ for every hyperedge $p \in P$, and
- a connection $\left\langle \Sigma_c, \Lambda_c, \left(\Sigma_c \xleftarrow{\iota^c_x} \Sigma^c_x \xrightarrow{\mu^c_x} \Sigma_x \right)_{x \in \gamma_c} \right\rangle$ for every hyperedge $c \in C$.

▶ **Example 8.** By putting together the process and the connection presented in Examples 2 and 5, we obtain the ARN JourneyPlanner depicted in Figure 1. Its underlying hypergraph consists of the points JP_1, JP_2, R_1 and R_2, the computation hyperedge JP, the communication hyperedge C, and the incidence map γ given by $\gamma_{\mathsf{JP}} = \{\mathsf{JP}_1, \mathsf{JP}_2\}$ and $\gamma_{\mathsf{C}} = \{\mathsf{JP}_2, \mathsf{R}_1, \mathsf{R}_2\}$.

An *interaction-point* of a $\langle \mathbb{S}\mathrm{ig}, L \rangle$-ARN A is a point of A that is not bound to both computation and communication hyperedges. We distinguish between requires-points and provides-points, as follows.

▶ **Definition 9** (Requires and Provides-point)**.** A *requires-point* of $\langle \mathbb{S}\mathrm{ig}, L \rangle$-ARN A is a point of A that is incident only with a communication hyperedge. Similarly, a *provides-point* of A is a point incident only with a computation hyperedge.

Morphisms of $\langle \mathbb{S}\mathrm{ig}, L\rangle$-ARNs can be defined as injective homomorphisms between their underlying hypergraphs that preserve all labels, except those associated with requires-points.

▶ **Definition 10** (Homomorphism of Hypergraphs). A *homomorphism* h between hypergraphs $\langle X, E, \gamma\rangle$ and $\langle X', E', \gamma'\rangle$ consists of functions $h^v\colon X \to X'$ and $h^e\colon E \to E'$, usually denoted simply by h, such that for any $x \in X$ and $e \in E$, $x \in \gamma_e$ if and only if $h^v(x) \in \gamma'_{h^e(e)}$.

▶ **Definition 11** (Morphism of ARNs). Given two $\langle \mathbb{S}\mathrm{ig}, L\rangle$-ARNs $A = \langle X, P, C, \gamma, \Sigma, \iota, \mu, \Lambda\rangle$ and $A' = \langle X', P', C', \gamma', \Sigma', \iota', \mu', \Lambda'\rangle$, a *morphism* $\varphi\colon A \to A'$ consists of

- an injective homomorphism $\varphi\colon \langle X, P, C, \gamma\rangle \to \langle X', P', C', \gamma'\rangle$ between the underlying hypergraphs of A and A' such that $\varphi(P) \subseteq P'$ and $\varphi(C) \subseteq C'$, and
- a family φ^{pt} of signature morphisms $\varphi^{pt}_x\colon \Sigma_x \to \Sigma'_{\varphi(x)}$, for $x \in X$,

such that

- for every non-requires-point $x \in X$, $\varphi^{pt}_x = 1_{\Sigma_x}$,
- for every hyperedge $p \in P$, $\Sigma_p = \Sigma'_{\varphi(p)}$, $\Lambda_p = \Lambda'_{\varphi(p)}$, and $\iota^p_x = (\iota')^{\varphi(p)}_{\varphi(x)}$ for any point $x \in \gamma_p$,
- for every hyperedge $c \in C$, $\Sigma_c = \Sigma'_{\varphi(c)}$, $\Lambda_c = \Lambda'_{\varphi(c)}$ and the following diagram is well-defined and commutative, for any point $x \in \gamma_c$.

$$\begin{array}{ccccc}
\Sigma_c & \xleftarrow{\iota^c_x} & \Sigma^c_x & \xrightarrow{\mu^c_x} & \Sigma_x \\
\downarrow{\scriptstyle 1_{\Sigma_c}} & & \downarrow{\scriptstyle 1_{\Sigma^c_x}} & & \downarrow{\scriptstyle \varphi^{pt}_x} \\
\Sigma'_{\varphi(c)} & \xleftarrow[(\iota')^{\varphi(c)}_{\varphi(x)}]{} & (\Sigma')^{\varphi(c)}_{\varphi(x)} & \xrightarrow[(\mu')^{\varphi(c)}_{\varphi(x)}]{} & \Sigma'_{\varphi(x)}
\end{array}$$

▶ **Proposition 12.** *The morphisms of $\langle \mathbb{S}\mathrm{ig}, L\rangle$-ARNs form a category denoted $\langle \mathbb{S}\mathrm{ig}, L\rangle$-$\mathbb{ARN}$.*

3 An institution of ARNs

We now turn our attention to the main contribution of the paper: the presentation of a logical framework for service orchestrations in an institutional setting. Similarly to the construction detailed in the previous section, the concepts discussed here are parameterised over an arbitrary logical system that satisfies a number of additional properties. More precisely, we consider a fixed institution $\mathcal{I} = \langle \mathbb{S}\mathrm{ig}^{\mathcal{I}}, \mathsf{Sen}^{\mathcal{I}}, \mathsf{Mod}^{\mathcal{I}}, \models^{\mathcal{I}}\rangle$ such that

- the category $\mathbb{S}\mathrm{ig}^{\mathcal{I}}$ of $\mathcal{I}$-signatures is cocomplete,
- there exist cofree models along any signature morphism, i.e. the reduct functor $\mathsf{Mod}^{\mathcal{I}}(\varphi)$ of any signature morphism φ admits a right adjoint, and
- the category of models $\mathsf{Mod}^{\mathcal{I}}(\Sigma)$ of any signature Σ has products.

An example of such institution is MA-LTL – a variant of linear temporal logic whose models are not traces, but Muller automata [11], and in which an automaton satisfies a sentence if and only if every trace accepted by the automaton satisfies the considered sentence. MA-LTL was originally proposed in [3] as an alternative to conventional linear temporal logic with the aim of capturing a more operational notion of service orchestration.

3.1 Signatures

We start by defining the category $\mathbb{S}\mathrm{pec}$-$\mathbb{ARN}_{\mathcal{I}}$ of signatures and signature morphisms of our institution, which we denote $\mathrm{SOC}_{\mathcal{I}}$. These are asynchronous relational networks determined

by the category $\mathbb{S}\mathrm{ig}^{\mathcal{I}}$ of signatures and by the labelling functor $\mathsf{Spec}^{\mathcal{I}}\colon |\mathbb{S}\mathrm{ig}^{\mathcal{I}}| \to \mathbb{C}\mathrm{at}$ that assigns to every $\mathcal{I}$-signature Σ the preorder category of Σ-presentations.

$$\mathbb{S}\text{pec-}\mathbb{ARN}_{\mathcal{I}} = \langle \mathbb{S}\mathrm{ig}^{\mathcal{I}}, \mathsf{Spec}^{\mathcal{I}} \rangle\text{-}\mathbb{ARN}$$

3.2 Sentences and Sentence Translations

The sentences of $\underline{\mathrm{SOC}}_{\mathcal{I}}$ express properties about the points of the considered ARNs.

▶ **Definition 13** (Sentence). For any network $A \in |\mathbb{S}\text{pec-}\mathbb{ARN}_{\mathcal{I}}|$, i.e. for any $\underline{\mathrm{SOC}}_{\mathcal{I}}$-signature A, the set $\mathsf{Sen}^{\mathrm{SOC}}_{\mathcal{I}}(A)$ of *(atomic) A-sentences* is defined as the set of pairs $\langle x, \rho \rangle$, usually denoted $@_x\, \rho$, where x is a point of A and ρ is an $\mathcal{I}$-sentence over Σ_x.

The *translation* of sentences is straightforward: for every morphism $\varphi\colon A \to A'$ in $\mathbb{S}\text{pec-}\mathbb{ARN}_{\mathcal{I}}$, the map $\mathsf{Sen}^{\mathrm{SOC}}_{\mathcal{I}}(\varphi)\colon \mathsf{Sen}^{\mathrm{SOC}}_{\mathcal{I}}(A) \to \mathsf{Sen}^{\mathrm{SOC}}_{\mathcal{I}}(A')$ is given by

$$\mathsf{Sen}^{\mathrm{SOC}}_{\mathcal{I}}(\varphi)(@_x\, \rho) = @_{\varphi(x)}\, \mathsf{Sen}^{\mathcal{I}}(\varphi^{pt}_x)(\rho)$$

for any point x of A and any $\mathcal{I}$-sentence ρ over the signature of x.

▶ **Proposition 14.** $\mathsf{Sen}^{\mathrm{SOC}}_{\mathcal{I}}$ *is a functor* $\mathbb{S}\text{pec-}\mathbb{ARN}_{\mathcal{I}} \to \mathbb{S}\mathrm{et}$.

3.3 Models and Model Reductions

The model functor of our institution assigns appropriate models of the underlying institution to the computation and communication hyperedges of the considered ARNs, and ground networks to their requires-points. Formally, we first define

- $\mathbb{M}\text{od-}\mathbb{ARN}_{\mathcal{I}}$ as the category $\langle \mathbb{S}\mathrm{ig}^{\mathcal{I}}, |\mathsf{Mod}^{\mathcal{I}}| \rangle\text{-}\mathbb{ARN}$,
- $\mathbb{M}\text{od-}\mathbb{ARN}^{A}_{\mathcal{I}}$, for any ARN $A = \langle X, P, C, \gamma, \Sigma, \iota, \mu, \Lambda^{spec} \rangle \in |\mathbb{S}\text{pec-}\mathbb{ARN}_{\mathcal{I}}|$, as the discrete subcategory of $\mathbb{M}\text{od-}\mathbb{ARN}_{\mathcal{I}}$ given by networks $\alpha = \langle X, P, C, \gamma, \Sigma, \iota, \mu, \Lambda^{mod} \rangle$ such that $\Lambda^{mod}_e \models^{\mathcal{I}}_{\Sigma_e} \Lambda^{spec}_e$ for every computation or communication hyperedge e of A, and
- $\mathbb{G}\mathbb{ARN}_{\mathcal{I}}$ as the full subcategory of $\mathbb{M}\text{od-}\mathbb{ARN}_{\mathcal{I}}$ determined by *ground* networks, i.e. by networks with no requires-points.

▶ **Definition 15** (Model). For any ARN $A \in |\mathbb{S}\text{pec-}\mathbb{ARN}_{\mathcal{I}}|$, the category $\mathsf{Mod}^{\mathrm{SOC}}_{\mathcal{I}}(A)$ of *A-models* or *A-interpretations* is defined as the comma category $\mathbb{M}\text{od-}\mathbb{ARN}^{A}_{\mathcal{I}} / \mathbb{G}\mathbb{ARN}_{\mathcal{I}}$.

It follows that A-interpretations are morphisms of ARNs $\nu\colon \alpha \to \beta$ such that $\alpha \in |\mathbb{M}\text{od-}\mathbb{ARN}^{A}_{\mathcal{I}}|$ and $\beta \in |\mathbb{G}\mathbb{ARN}_{\mathcal{I}}|$, which can also be seen as collections of ground networks that are designated to the requires-points of α. In order to explain this in more detail let us introduce the following notions of dependency and ARN defined by a point.

▶ **Definition 16** (Dependency). Let x and y be points of an ARN α. The point x is said to be *dependent* on y if there exists a path from x to y that begins with a computation hyperedge, i.e. if there exists an alternating sequence $x\, e_1\, x_1 \cdots e_n\, y$ of (distinct) points and hyperedges such that $x \in \gamma_{e_1}$, $y \in \gamma_{e_n}$, $x_i \in \gamma_{e_i} \cap \gamma_{e_{i+1}}$ for any $1 \leq i < n$, and $e_1 \in P$.

▶ **Definition 17** (ARN Defined by a Point). The *sub-ARN defined by a point* x of an ARN α is the full sub-ARN α_x of α determined by x and the points on which x is dependent.

One can now see that any interpretation $\nu\colon \alpha \to \beta$ of an ARN $A \in |\mathbb{S}\text{pec-}\mathbb{ARN}_{\mathcal{I}}|$ assigns to each requires-point x of α the ground sub-ARN $\beta_{\nu(x)}$ of β defined by $\nu(x)$.

With respect to model reducts, one can easily see that every ARN morphism $\varphi\colon A \to A'$ in $\mathbb{S}\text{pec-}\mathbb{ARN}_{\mathcal{I}}$ and every network $\alpha' \in |\mathbb{M}\text{od-}\mathbb{ARN}^{A'}_{\mathcal{I}}|$ determine a (unique) morphism of ARNs

$\varphi\colon \alpha \to \alpha'$ in $\mathbb{M}\text{od-}\mathbb{ARN}_{\mathcal{I}}$ such that $\alpha \in |\mathbb{M}\text{od-}\mathbb{ARN}^{A}_{\mathcal{I}}|$. This observation allows us to define the reduction of interpretations as the left composition with the considered ARN morphism. Hence, for every ARN morphism $\varphi\colon A \to A'$, $\mathsf{Mod}^{\mathrm{SOC}}_{\mathcal{I}}(\varphi)$ is given by $\mathsf{Mod}^{\mathrm{SOC}}_{\mathcal{I}}(\varphi)(\nu') = \varphi;\nu'$ for A'-interpretations ν' and $\mathsf{Mod}^{\mathrm{SOC}}_{\mathcal{I}}(\varphi)(\zeta') = \zeta'$ for A'-interpretation homomorphisms ζ'.

▶ **Proposition 18.** $\mathsf{Mod}^{\mathrm{SOC}}_{\mathcal{I}}$ *is a contravariant functor* $\mathbb{S}\text{pec-}\mathbb{ARN}^{op}_{\mathcal{I}} \to \mathbb{C}\text{at}$.

3.4 The Satisfaction Relation

The evaluation of $\underline{\mathrm{SOC}}_{\mathcal{I}}$ sentences with respect to the interpretations relies on the concepts of diagram of a network and of model defined by a point, whose purpose is to describe the observable behaviour of a ground network through one of its points.

▶ Fact 19 (Diagram of an ARN). Every ARN $\alpha = \langle X, P, C, \gamma, \Sigma, \iota, \mu, \Lambda^{mod}\rangle \in |\mathbb{M}\text{od-}\mathbb{ARN}_{\mathcal{I}}|$ defines a diagram $D_\alpha\colon \mathbb{J}_\alpha \to \mathbb{S}\text{ig}^{\mathcal{I}}$ as follows:

– $\mathbb{J}_\alpha$ is the free preorder category given by the set of objects

$$X \cup P \cup C \cup \{\langle c, x, \alpha\rangle \mid c \in C, x \in \gamma_c\}$$

and the arrows

- $\{x \to p \mid p \in P, x \in \gamma_p\}$ for computation hyperedges, and
- $\{c \leftarrow \langle c, x, \alpha\rangle \to x \mid c \in C, x \in \gamma_c\}$ for communication hyperedges;

– D_α is the functor that provides the signatures of ports, processes and channels, together with the appropriate mappings between them.

Since $\mathbb{S}\text{ig}^{\mathcal{I}}$ is cocomplete, we can define the signature of a network based on its diagram.

▶ **Definition 20** (Signature of an ARN)**.** The signature of an ARN $\alpha \in |\mathbb{M}\text{od-}\mathbb{ARN}_{\mathcal{I}}|$ is the colimiting cocone $\xi\colon D_\alpha \Rightarrow \Sigma_\alpha$ of the diagram D_α.

The most important construction that allows us to define the satisfaction relation is the one that defines the observed behaviour of a (ground) network at one of its points.

▶ **Definition 21** (Model Defined by a Point)**.** Let x be a point of a ground ARN $\beta \in |\mathbb{G}\mathbb{ARN}_{\mathcal{I}}|$. The *observed model* Λ^{mod}_x at x is given by the reduct $\mathsf{Mod}^{\mathcal{I}}(\xi_x)(\Lambda^{mod}_{\beta_x})$, where

– $\beta_x = \langle X, P, C, \gamma, \Sigma, \iota, \mu, \Lambda^{mod}\rangle$ is the sub-ARN of β defined by x,
– $\xi\colon D_{\beta_x} \Rightarrow \Sigma_{\beta_x}$ is the signature of β_x,
– $\Lambda^{mod}_{\beta_x}$ is the product $\prod_{e \in P \cup C} \Lambda^{mod}_{\beta_x, e}$, and
– $\Lambda^{mod}_{\beta_x, e}$ is the cofree expansion of Λ^{mod}_e along ξ_e, for any hyperedge $e \in P \cup C$.

We now have all the necessary concepts for defining the satisfaction of $\underline{\mathrm{SOC}}_{\mathcal{I}}$ sentences by interpretations. Let us thus consider an ARN $A \in |\mathbb{S}\text{pec-}\mathbb{ARN}_{\mathcal{I}}|$, an A-interpretation $\nu\colon \alpha \to \beta$ and an A-sentence $@_x\,\rho$. Then

$$\nu \models^{\mathrm{SOC}}_{\mathcal{I},A} @_x\,\rho \quad \text{if and only if} \quad \mathsf{Mod}^{\mathcal{I}}(\nu^{pt}_x)\big(\Lambda^{mod}_{\nu(x)}\big) \models^{\mathcal{I}}_{\Sigma_x} \rho,$$

where $\Lambda^{mod}_{\nu(x)}$ is the observed model at $\nu(x)$ in β.

The construction of the institution of ARNs is completed by the following result, which states that satisfaction is invariant with respect to changes of ARNs.

▶ **Proposition 22.** *For every ARN morphism* $\varphi\colon A \to A'$ *in* $\mathbb{S}\text{pec-}\mathbb{ARN}_{\mathcal{I}}$*, any* A'*-interpretation* ν' *and any* A*-sentence* $@_x\,\rho$*,*

$$\nu' \models^{\mathrm{SOC}}_{\mathcal{I},A'} \mathsf{Sen}^{\mathrm{SOC}}_{\mathcal{I}}(\varphi)(@_x\,\rho) \quad \text{if and only if} \quad \mathsf{Mod}^{\mathrm{SOC}}_{\mathcal{I}}(\varphi)(\nu') \models^{\mathrm{SOC}}_{\mathcal{I},A} @_x\,\rho.$$

▶ **Corollary 23.** $\underline{\mathrm{SOC}}_{\mathcal{I}} = \big\langle \mathbb{S}\text{pec-}\mathbb{ARN}_{\mathcal{I}}, \mathsf{Sen}^{\mathrm{SOC}}_{\mathcal{I}}, \mathsf{Mod}^{\mathrm{SOC}}_{\mathcal{I}}, \models^{\mathrm{SOC}}_{\mathcal{I}}\big\rangle$ *is an institution.*

4 Conclusions

In this paper we proposed a logical framework of networks of processes that can be used in combination with concepts and results specific to the logic-programming paradigm [3] to offer an integrated semantics for the static and dynamic aspects of service-oriented computing. We distinguish between specifications of services and their models, which are orchestrations of components that rely upon externally provided services. The resulting institution is parameterised over an arbitrary logical system such that (a) its category of signatures is cocomplete, (b) there exist cofree models along any signature morphism, (c) the category of models of any signature has products. This level of generality encourages us to further investigate the service-oriented computing paradigm over a variety of logics. An issue to be pursued towards this heterogeneous setting is the way in which the change of the underlying logics induces appropriate translations between the corresponding institutions of ARNs.

Acknowledgements. The author would like to thank José Fiadeiro for many useful discussions and feedback on the logic-programming semantics of services. This research has been supported by a grant of the Romanian National Authority for Scientific Research, CNCS-UEFISCDI, project number PN-II-ID-PCE-2011-3-0439.

References

1 Daniel Brand and Pitro Zafiropulo. On communicating finite-state machines. *Journal of the ACM*, 30(2):323–342, 1983.

2 Roberto Bruni, Fabio Gadducci, and Alberto Lluch-Lafuente. A graph syntax for processes and services. In Cosimo Laneve and Jianwen Su, editors, *Web Services and Formal Methods*, Lecture Notes in Computer Science, pages 46–60. Springer, 2009.

3 Ionuţ Ţuţu and José L. Fiadeiro. A logic-programming semantics of services. In Reiko Heckel and Stefan Milius, editors, *Conference on Algebra and Coalgebra in Computer Science*, Lecture Notes in Computer Science, pages 299–313. Springer, 2013.

4 Răzvan Diaconescu. *Institution-independent model theory*. Studies in Universal Logic. Birkhäuser, 2008.

5 Gian Luigi Ferrari, Dan Hirsch, Ivan Lanese, Ugo Montanari, and Emilio Tuosto. Synchronised hyperedge replacement as a model for service oriented computing. In Frank de Boer, Marcello Bonsangue, Susanne Graf, and Willem de Roever, editors, *Formal Methods for Components and Objects*, Lecture Notes in Computer Science, pages 22–43. Springer, 2005.

6 José L. Fiadeiro and Antónia Lopes. An interface theory for service-oriented design. In Dimitra Giannakopoulou and Fernando Orejas, editors, *Fundamental Approaches to Software Engineering*, Lecture Notes in Computer Science, pages 18–33. Springer, 2011.

7 José L. Fiadeiro and Antónia Lopes. A model for dynamic reconfiguration in service-oriented architectures. *Software and Systems Modeling*, pages 1–19, 2012.

8 José L. Fiadeiro, Antónia Lopes, and Laura Bocchi. Algebraic semantics of service component modules. In José L. Fiadeiro and Pierre-Yves Schobbens, editors, *Workshop on Algebraic Development Techniques*, Lecture Notes in Computer Science, pages 37–55. Springer, 2006.

9 José L. Fiadeiro, Antónia Lopes, and Laura Bocchi. An abstract model of service discovery and binding. *Formal Aspects of Computing*, 23(4):433–463, 2011.

10 Joseph A. Goguen and Rod M. Burstall. Institutions: Abstract model theory for specification and programming. *Journal of the ACM*, 39(1):95–146, 1992.

11 Dominique Perrin and Jean Éric Pin. *Infinite Words: Automata, Semigroups, Logic and Games*. Pure and Applied Mathematics. Elsevier Science, 2004.